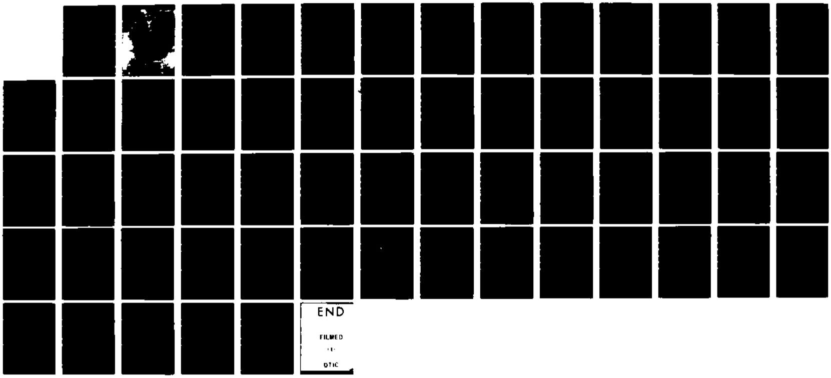
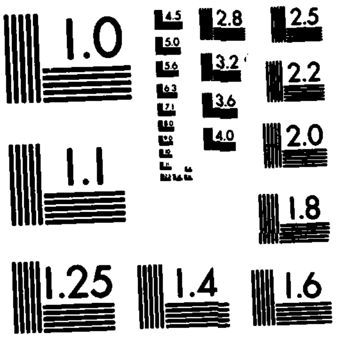
AD-A124 742 IMPLEMENTATION OF A GRID-BASED LINE-DRAWING QUANTIZATION SYSTEM(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 1/1

UNCLASSIFIED J N PEERY DEC 82 AFIT/GE/EE/82D-53 F/G 12/1 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0

AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE

IMPLEMENTATION OF A GRID-BASED LINE-DRAWING QUANTIZATION SYSTEM

THESIS

AFIT/GE/EE/82D-53

Jerry N. Peery
2d Lt USAF

SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/GE/EE/82D-53

2. GOVT ACCESSION NO.
AD-A124 742

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
IMPLEMENTATION OF A
GRID-BASED LINE-DRAWING
QUANTIZATION SYSTEM

5. TYPE OF REPORT & PERIOD COVERED
M.S. Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Jerry N. Peery
2d Lt USAF

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS (AFIT/EN)
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS (AFIT/EN)
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

12. REPORT DATE
December 1982

13. NUMBER OF PAGES
56

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.
Lynn Wolaver
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

4 JAN 1983

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Grid INtersect Codes
Line-Drawings
Digitizer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of this thesis is to develop a system that can be used to implement and evaluate various grid intersect codes. Grid intersect codes are used in line-drawing quantization schemes which provide specialized methods for encodlng drawings which consist soley of thin lines on a contrasting background. Examples of such drawings include weather maps, blueprints, and English text. The system will consist of a NOVA 1200 computer and digitizer.

The Digitizer consists of a sketch pad, a pen, and the electrical circuitry to determine the location of the pen relative to the sketch pad. The position of the pen is converted to digital information that is used as the input to the computer. The computer can use this data to implement various grid intersect algorithms. This thesis discusses the design of the interface of these separate components into a system.

0

IMPLEMENTATION OF A GRID-BASED LINE-DRAWING QUANTIZATION SYSTEM

THESIS

Jerry N. Peery
AFIT/GE/EE/82D-53 2d Lt USAF

AFIT/GE/EE/82D-53

IMPLEMENTATION OF A GRID-BASED LINE-DRAWING QUANTIZATION SYSTEM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Jerry N. Peery
2d Lt USAF

Graduate Electrical Engineering

December 1982

Approved for public release; distribution unlimited.

Acknowledgments

I would like to thank Major Kenneth Castor for sponsoring this work. I am grateful to him for his advice, patience, and assistance during this project. Also, I would like to thank the lab technicians for their assistance and advice in obtaining parts for this thesis. I am extremely grateful to my sister-in-law, Anita, for providing the typewriter used to type this thesis. Finally, I would like to thank my family, Renee, Michael, and Teresa for giving me the time and understanding needed during this period.

Jerry N. Peery

Contents

List of Figures

List of Tables

Abstract

The object of this thesis is to develop a system that can be used to implement and evaluate various grid intersect codes. Grid intersect codes are used in line-drawing quantization schemes which provide specialized methods for encoding drawings which consist solely of thin lines on a contrasting background. Examples of such drawings include weather maps, blue prints, and English text. The system will consist of a NOVA 1200 computer and digitizer. The digitizer consists of a sketch pad, a pen, and the electrical circuitry to determine the location of the pen relative to the sketch pad. The position of the pen is converted to digital information that is used as the input to the computer. The computer can use this data to implement various grid intersect algorithms. This thesis discusses the design of the interface of these separate components into a system.

I Introduction

The objective of this thesis effort is the design and implementation of a grid-based line-drawing quantization system. Line drawing quantization schemes provide specialized methods for encoding drawings which consist solely of thin lines on a contrasting background. Examples of such drawings include weather maps, blueprints, and English text. Existing systems, such as the electronic blackboard systems, use simple grid-based quantization methods to digitize the source information for transmission over a communication network. Numerous quantization schemes have been devised which vary in complexity and resolution capability. The overall objective of this thesis project is to provide a system that can be used to evaluate and analyze these various quantization schemes.

The basic grid intersect code is relatively easy to implement. The grid consists of parallel lines in the horizontal (X) direction and parallel lines in the vertical (Y) direction. The basic grid intersect code represents a line drawn on the grid as a collection of points (X,Y coordinate pairs). The X,Y coordinate pair closest to the point where the drawn line intersects a X or Y grid line is chosen to represent that point on the line. It is obvious that this results in a degree of error. The more closely spaced the grid lines are together the less the error.

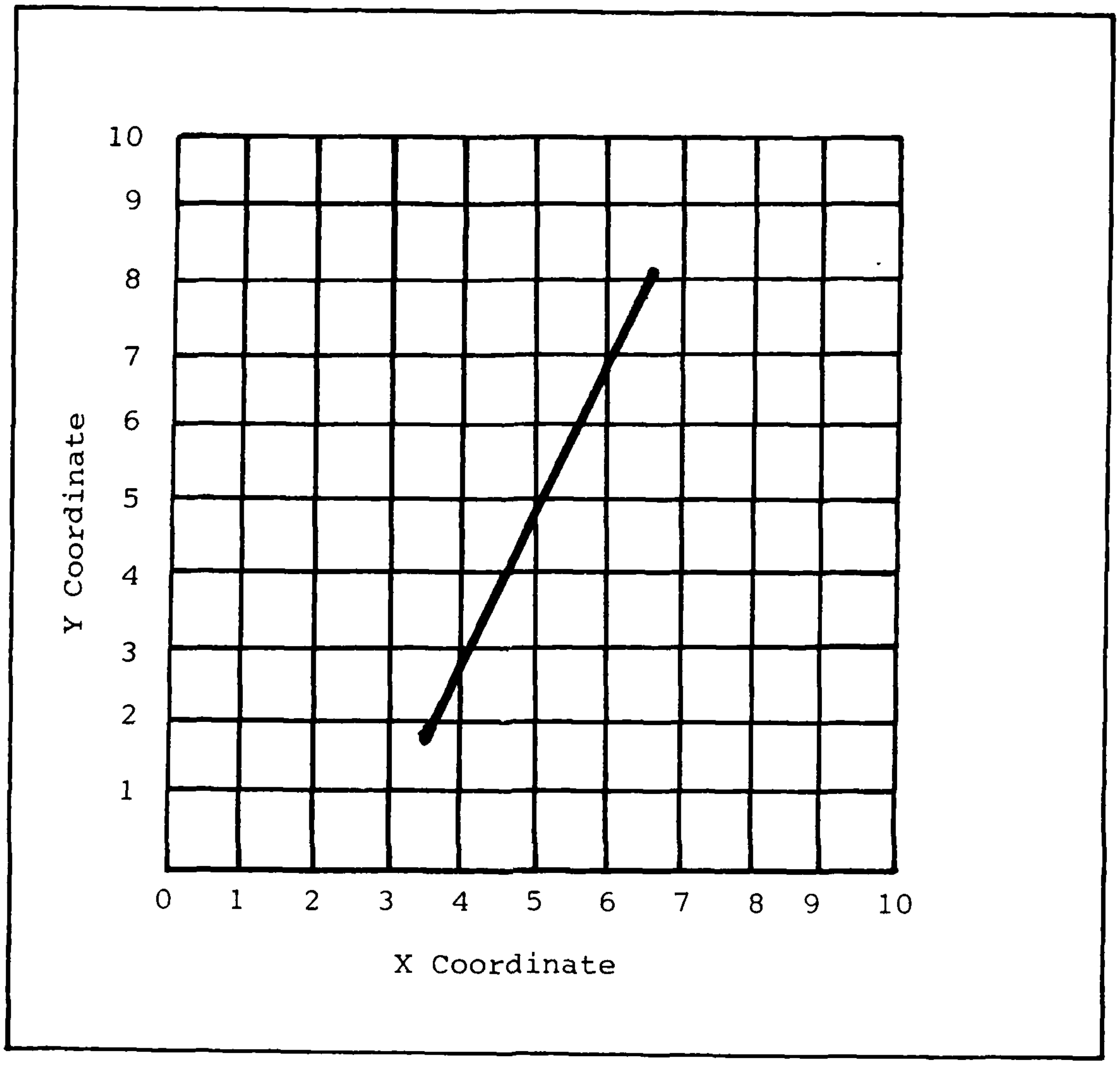


Fig. 1   Example of Grid and Drawn Line

The concept of quantizing a line using a grid-based scheme will be illustrated using the simple grid intersect code. The line drawn on the grid in Figure 1 can be quantized to the points in Table 1. The line can then be represented by these data points. The line can be reconstructed from these data points as shown in Figure 2.

Table I

Coordinate Representation of Line

Data Point	Coordinates (X,Y)	Digital Representation (X,Y)
1	4,2	0100,0010
2	4,3	0100,0011
3	4,3	0100,0011
4	5,4	0101,0100
5	5,5	0101,0101
6	5,5	0101,0101
7	6,6	0110,0110
8	6,7	0110,0111
9	6,7	0110,0111
10	7,8	0111,1000

The error shown by this example is exaggerated because of the large spacing between grid lines. It can be seen that a reasonable representation of the line can be obtained even from this basic grid intersect code. A better representation of the line can be obtained by using one of the more complex grid intersect codes. The length of the digital words depend on the size of the grid. The greater the number of grid lines in the grid the larger the data

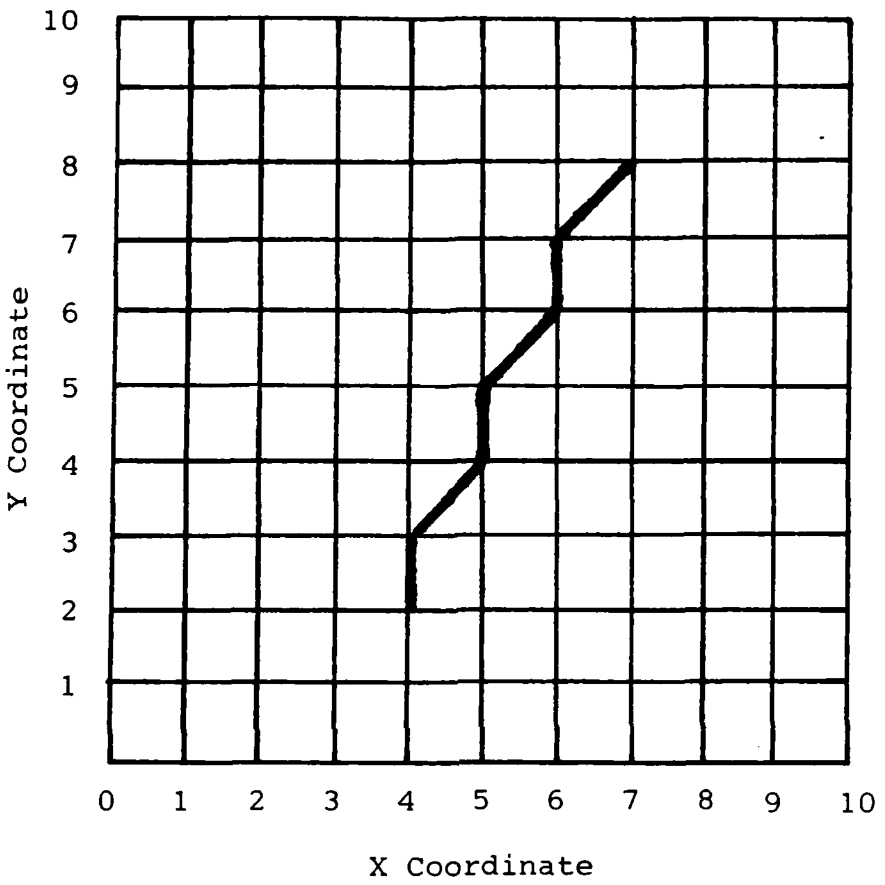


Fig. 2 Reconstructed Line

word needed to represent a data point. The number of bits required in the data word can be determined from the expression

$$L = \log_2 N$$

where L is the length of the data word in bits, and N is the number of grid lines in a coordinate direction (X or Y).

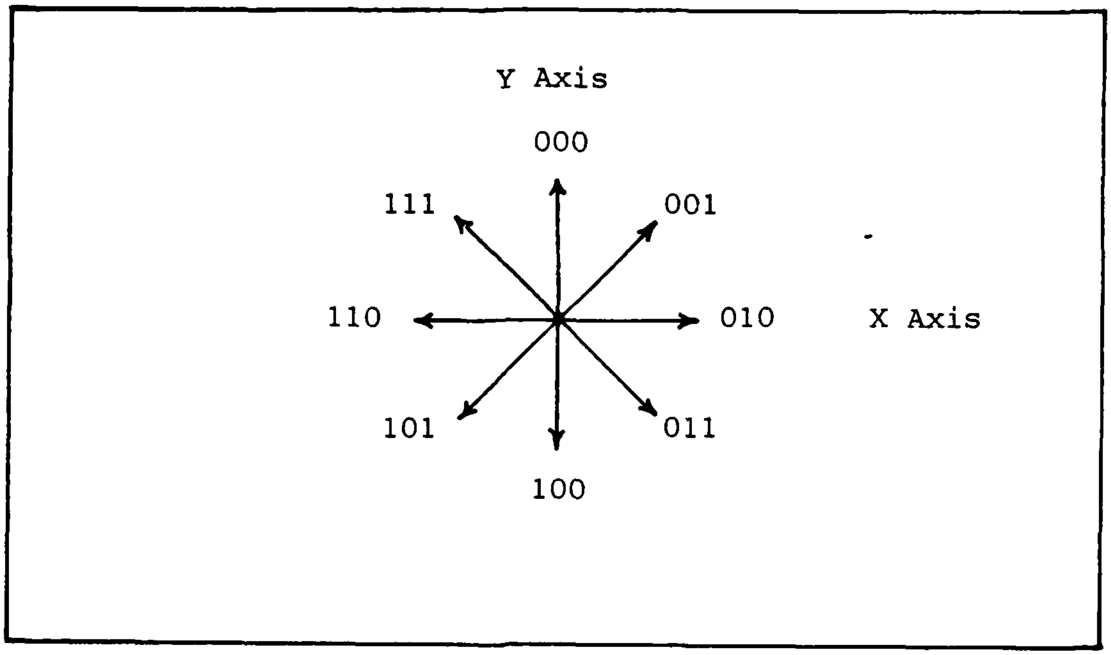


Fig. 3 Digital Representation of Directions

Although this method does quantize the data for digital representation, the data words can be quite long and cumbersome to work with. A more efficient method would be to give the coordinates of the first data point to be quantized, and thereafter giving a series of directions to proceed to reconstruct the line. From a grid data point there are eight directions to proceed to arrive at a new data point that is less than the distance between two grid lines. These directions are shown in Figure 3. Eight directions can be represented by eight digital data words 3 bits long. Therefore, a line can be represented by giving the coordinates of the first data point and thereafter giving a series of directions. The directions can be represented by

Table II

Direction Representation of Line

Data Point	Coordinates (X,Y)	Direction from Preceeding Data
1	4,2	-
2	4,3	000
3	4,3	-
4	5,4	001
5	5,5	000
6	5,5	-
7	6,6	001
8	6,7	000
9	6,7	-
10	7,8	001

eight 3 bit data words regardless of the size involved. This is much more efficient and is faster and less cumbersome to work with. The example line in Figure 2 can be represented by the digital words given in Table II. As can be seen when compared to Table I this representation requires less bits and therefore is faster to transmit serially.

The thrust of this thesis is to interface a Talos System parallel binary output board (electronic sketch pad) to a Data General computer system. The system resulting from this thesis will be used to implement and evaluate various grid intersect codes. The design, implementation, and testing of grid intersect codes is a topic suitable for follow on theses.

This thesis consists of designing and implementing four major sections; a parallel to serial converter, a serial to parallel converter, an interface to the computer, and the software to load the data into the computer.

The parallel to serial converter makes the output of the Talos parallel output (digitizer) compatible to RS-232C specifications. This makes the remote placement of the digitizer more economical as fewer land lines are required. The serial to parallel converter changes the data back to parallel form in order that it may be loaded into the computer.

The interface to the computer places the data onto the data bus of the computer at the appropriate time. The timing of loading data on the data bus is critical. Therefore; the circuitry that allows communication between the interface and the computer comprises the majority of the interface.

The software used to load the data into the computer from the interface is the last section to be implemented. The software tests the interface to determine when a data word is available, loads the data into memory, and manipulates the data as required by the grid intersect algorithm. Manipulating data by grid intersect algorithms will not be addressed in this thesis.

A brief description of the digitizer and the computer will be followed by a more detailed description of the parallel to serial converter, the serial to parallel converter, the computer interface, and the software. The parallel to serial converter, the serial to parallel converter, and the computer interface will collectively be referred to as the interface.

II The Digitizer

The digitizer used is a Digi-Kit-Izer, part number 11910-2, made by Talos System Inc. of Scottsdale, Arizona. The digitizer consists of a sketch pad that can determine the relative position of a pen by means of magnetic coupling. The position of the pen is converted to digital information in the form of 16 bit parallel words. The digitizer contains four major sections; a multiplex board (sketch pad), a transducer (pen), a digital control board, and an output board.

The multiplex board is a surface area containing 64 parallel conductors in both the horizontal (X) and vertical (Y) direction. The centers of the conductors are spaced .2 inches apart. Each conductor is 12.8 inches long with margins of .9 inches on both ends. This forms a usable grid surface area of 11 X 11 inches. The grid is shown in Figure 4. The grid has a plastic cover that protects it and acts as a writing surface. Paper can be taped to this surface to form a sketch pad.

The pen is a transducer that is electrically connected to the digitizer. Inside the tip of the pen is a small coil that is driven by an AC amplifier at a frequency of 102.4 khz (Ref 6 :14). This AC signal creates a magnetic field that is used to determine the location of the pen.

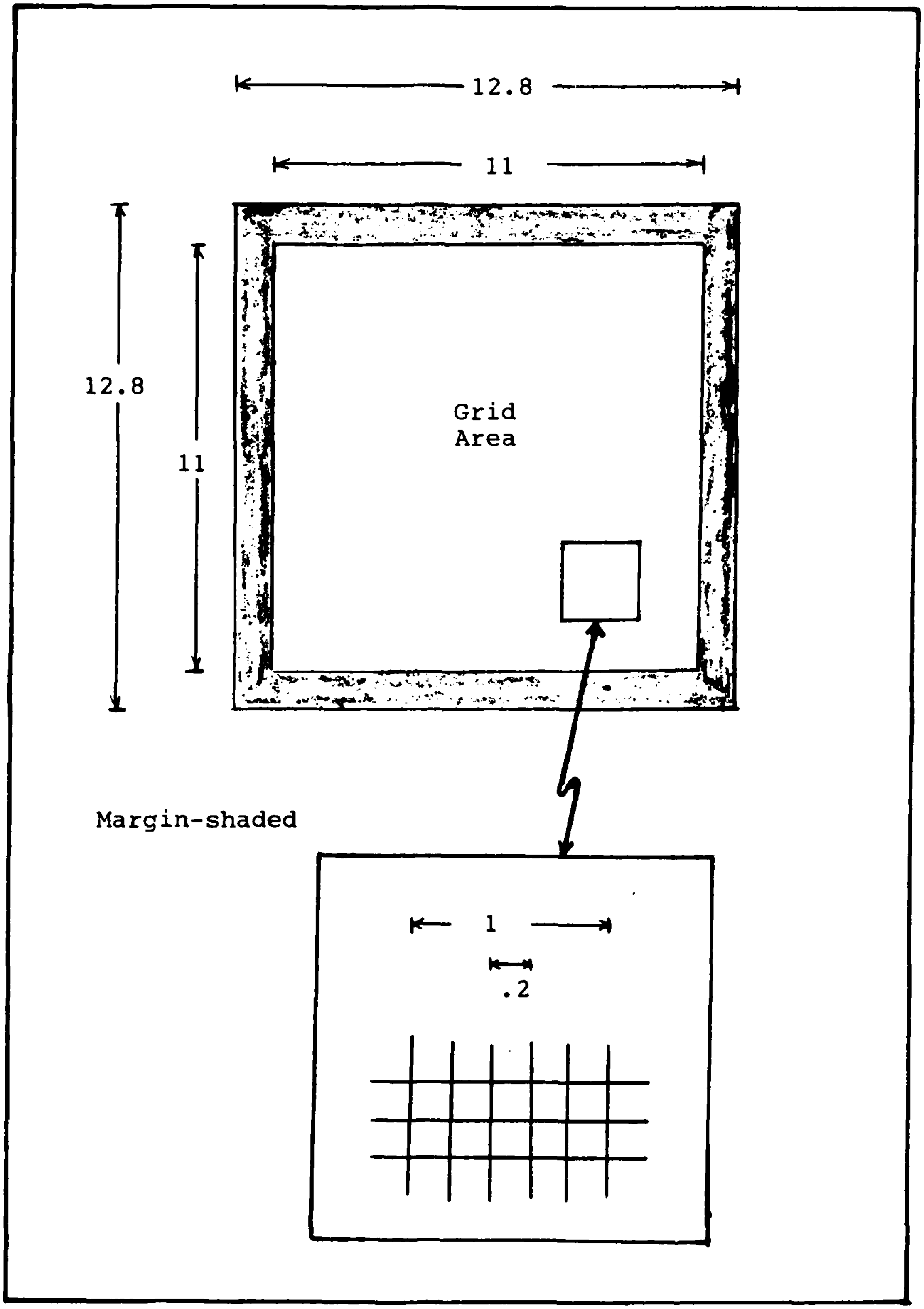


Fig. 4 Grid (measurements are in inches)

The pen is used in the same manner as a standard ball point pen. The pen contains a ball point pen ink filler cartridge that is easily replacable. As the pen is used to draw on the paper taped to the writing surface the position of the magnetic field is detected by the control board and quantized to digital data. The control board determines the relative position of the magnetic field by electrically scanning the multiplex board. The quantized digital data is arranged into the proper format by the output board. The output board places the data onto a 16 bit parallel data bus. The block diagram of the digitizer is shown in Figure 5.

Theory

The pen acts as the primary of a transformer, the conductors act as the secondary. When the pen is in close proximity to a conductor it induces a current in the conductor at the same frequency as the AC signal in the pen, but at a different phase angle. By comparing phase angles in different conductors the relative position of the pen can be determined. The analog circuitry that determines pen position consists of a preamplifier, a phase detector, an active filter, and a zero crossing detector. The output of this circuitry is a pulse that represents the change in phase angle.

A counter is used to determine the X and Y coordinates. The rate of the counter corresponds to how many

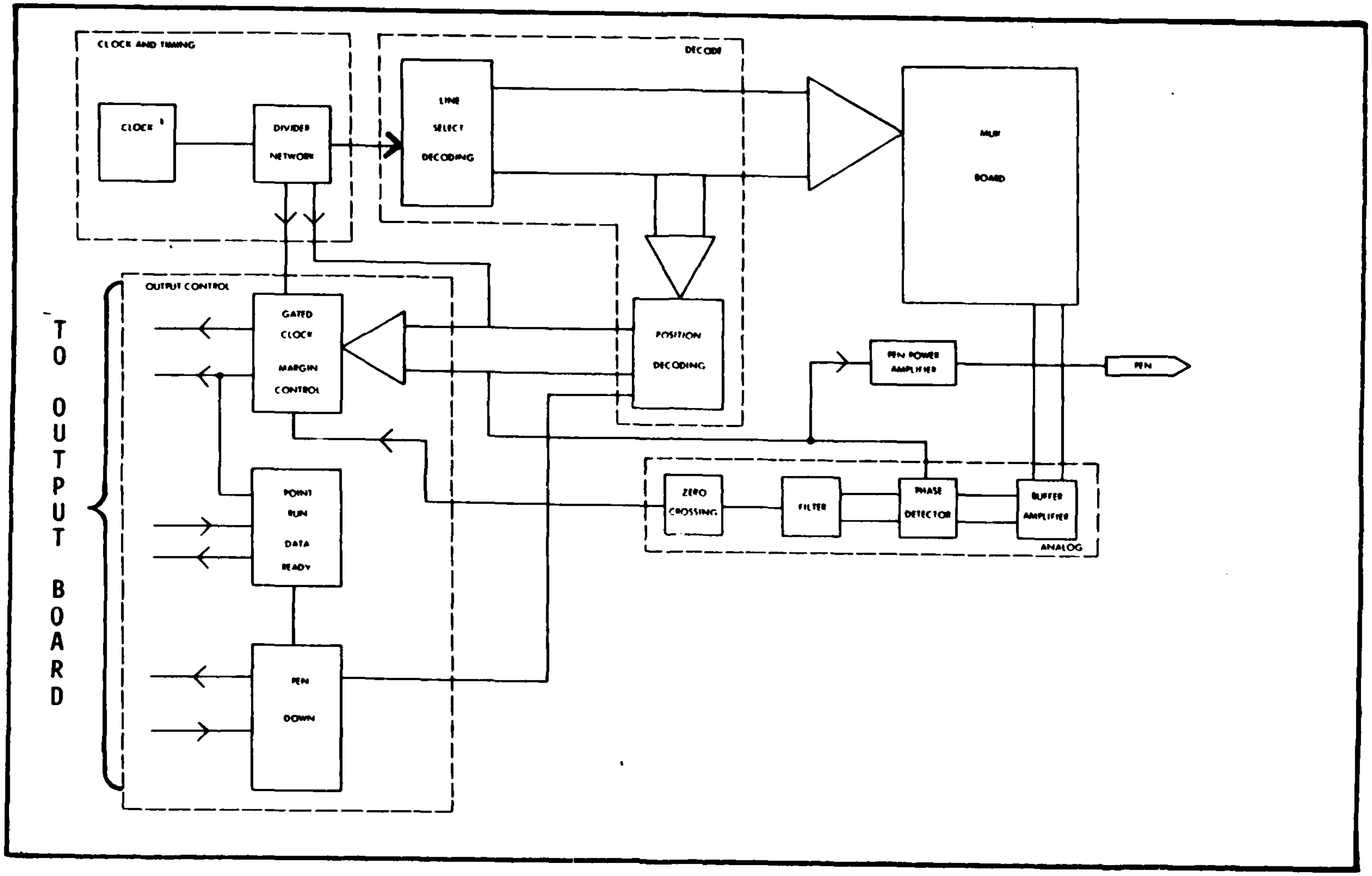


Fig. 5 Digitizer Block Diagram
(Ref 6 : 15)

conductor lines are sampled in a specific time. The rate used is 200 counts in the time it takes to sample 5 conductors (1 inch). The counter continues to count until it is stopped by the arrival of the pulse that is output by the zero crossing detector. The count in the counter represents the distance from the margin to the point at which the pen is located. This is done along both the X and the Y axis to form a X,Y coordinate pair. The data containing position information is transferred to the output board. The output board adds control information and loads the data onto the 16 bit data bus. The format of the data word is shown in Table III.

Output Interface

The output of the digitizer is a 16 bit parallel data word. The first 12 bits are position coordinates. The last 3 bits are control information used to describe this data. The output data bus is connected to the digitizer at connector J1. The connections of J1 are shown in Table IV. The data bus is connected to the computer interface at connector J2. The connections of J2 are the same as J1 reversed. For example; pin #26 of J1 corresponds to pin #1 of J2. Pin #26 does not exist for J2.

The digitizer output consists of a X data word transmission and then a Y word data transmission. The time required for a complete X,Y coordinate pair to be transmitted is 10 milliseconds.

Table III Parallel Data Format (Ref 7:7)

Bit Number	1st word	2nd Word
1	X_0 (LSB)	Y_0 (LSB)
2	X_1	Y_1
3	X_2	Y_2
4	X_3	Y_3
5	X_4	Y_4
6	X_5	Y_5
7	X_6	Y_6
8	X_7	Y_7
9	X_8	Y_8
10	X_9	Y_9
11	X_{10}	Y_{10}
12	X_{11} (MSB)	Y_{11} (MSB)
13	Not Used	Not Used
14	Pen Up/$\overline{\text{down}}$	Pen UP/$\overline{\text{down}}$
15	$X/\overline{Y} = 1$	$X/\overline{Y} = 0$
16	$\overline{\text{Margin-Fault}}$	$\overline{\text{Margin-Fault}}$

The Data Ready Strobe (DRS) pulses at the beginning of each data word (every 5 milliseconds). The timing diagram is shown in Figure 6.

Table IV Interface Connections (Ref 7:8)

J1 connector pin number	Functional description
1	Not used
2	Not used
3	Not used
4	$\overline{\text{Point}}$/Run
5	$\overline{\text{Data Ready Strobe}}$
6	X/$\overline{\text{Y}}$ Data Mode
7	Signal Ground
8	Not used
9	$\overline{\text{Margin Fault}}$
10	Stylus up/$\overline{\text{down}}$
11	Data 0
12	Data 1
13	Data 2
14	Data 3
15	Data 4
16	Data 5
17	Data 6
18	Data 7
19	Data 8
20	Data 9
21	Data 10
22	Data 11
23	Not used
24	Not used
25	15v Power Input
26	Power Ground Input

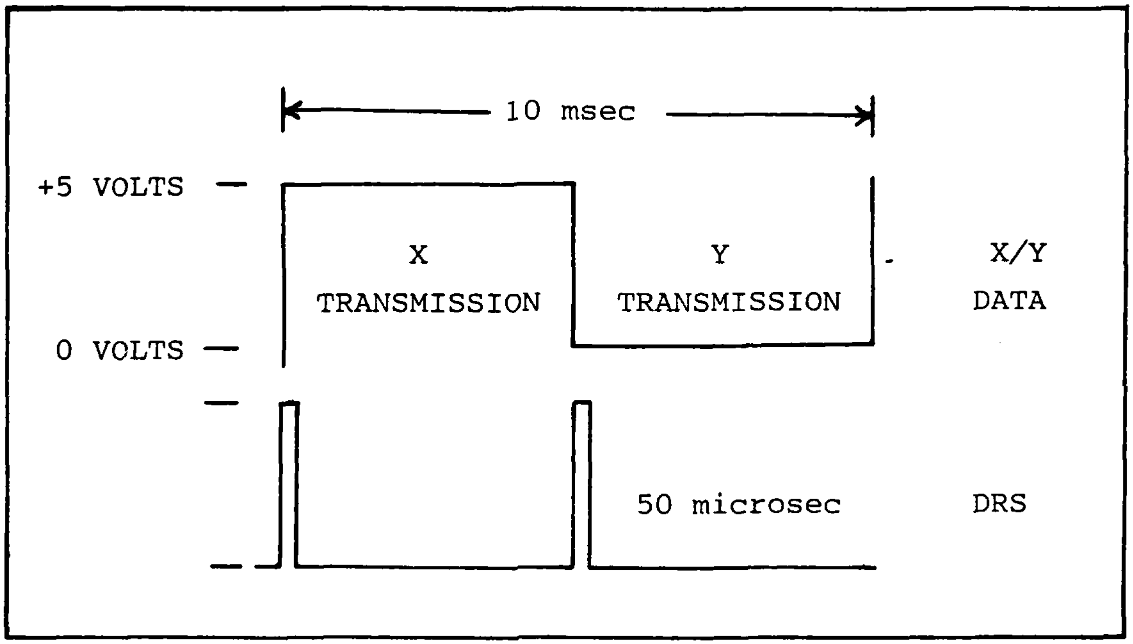


Fig. 6 Data Timing

II The Computer

The computer used in this project was a NOVA 1200 series computer made by Data General. This computer has a 16 bit word length. It has 32K words of random access memory (RAM). The central processing unit contains four accumulators. A teletype (TTY) was provided for use with the computer.

The best means available to program the computer is by means of data switches located on the front panel of the computer. The software needed to operate the interface was hand assembled and manually loaded into the computer. Since the amount of software required to operate the interface is relatively small this was not a major problem. In the future a floppy disk drive unit will be added to the computer to make it more efficient to use. This will be necessary because the programs needed to implement complex grid intersect codes will be too long to hand assemble and load.

Data Lines

The devices connected to the data bus of the NOVA 1200 computer are connected in a daisy chain fashion. The last device in the daisy chain must have the appropriate termination circuitry. This circuit is sold by Data General and is used to reduce ringing. The easiest way to obtain access to the data bus is to disconnect the data bus daisy chain and insert the interface device at that point. The arrangement of the daisy chained data bus after this has been performed is shown in Figure 7.

The data bus is routed into the interface on connector J5 and routed out on J6. The corresponding numbered pins of the two connectors are connected together by jumper wires. The data signals that are used in the interface are shown in Table V.

The input lines CLR and DATIA must have a filter circuit at the input of the interface circuit to improve noise margins. The filter circuit is shown in Figure 8.

The data from the interface is loaded into the computer via a programmed transfer. The programmed transfer signal timing diagram is shown in Figure 9. The computer enables the interface device by placing the device address on the data lines DS0-DS5. The processor then generates a DATIA pulse to place the data in the interface on the data bus. A strobe pulse is generated by the computer at the end of the DATIA pulse to load data into the accumulator. After the data has been loaded, a clear (CLR) pulse is generated by the computer to clear the interface device.

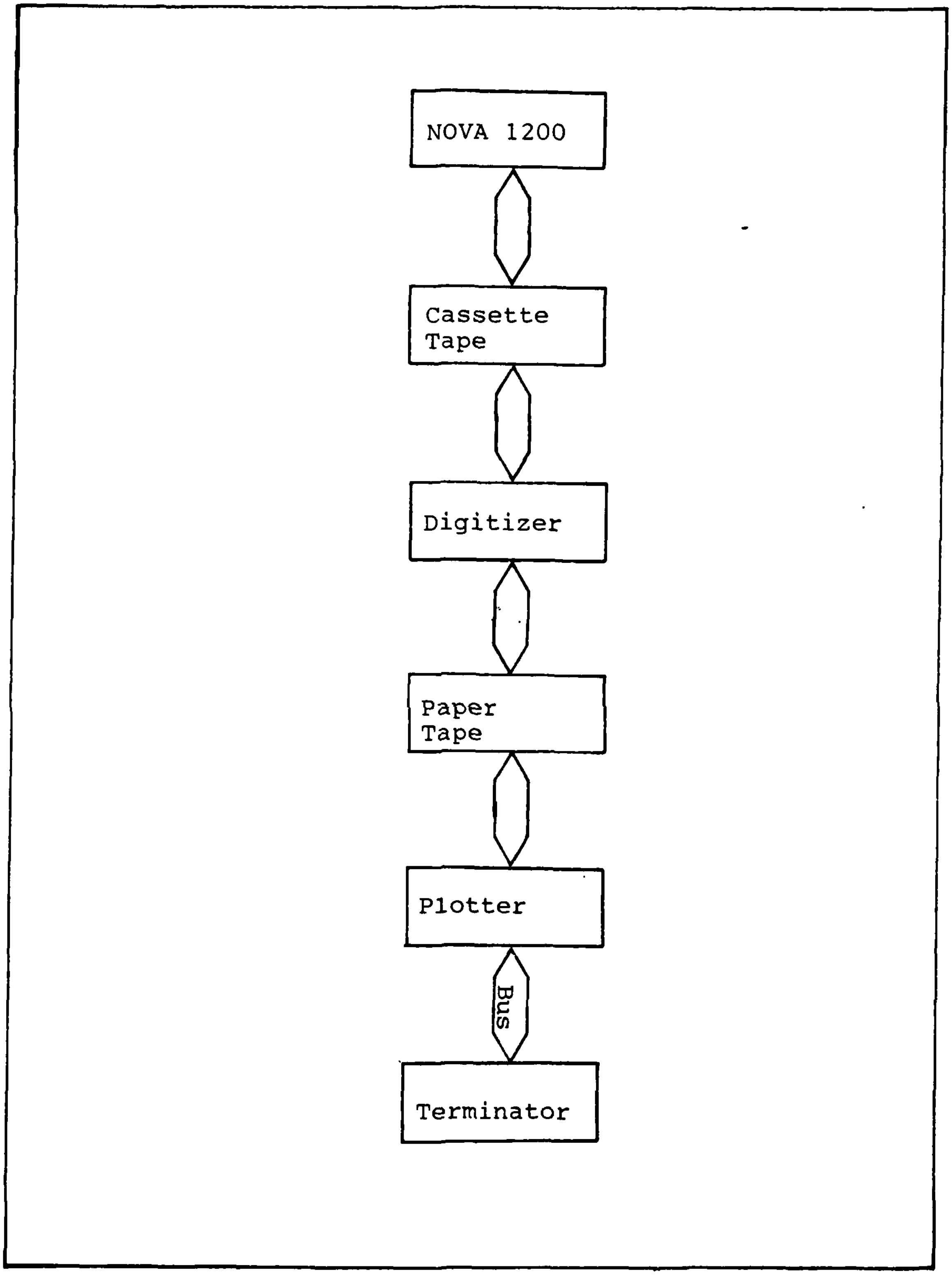


Fig. 7 NOVA 1200 Data Bus

Table V
In-Out Bus Signal Connections

Signal	Direction	Pins
CLR	D	2
DO	B	3
D1	B	4
D2	B	5
D3	B	6
D4	B	7
D5	B	8
D6	B	9
D7	B	10
D8	B	11
D9	B	12
D10	B	13
D11	B	14
D12	B	15
D13	B	16
D14	B	17
D15	B	18
DATIA	D	19
DS0	D	32
DS1	D	33
DS2	D	34
DS3	D	35
DS4	D	36
DS5	D	37
SELD	P	47

B	Bidirectional
D	From processor to device
P	From device to processor

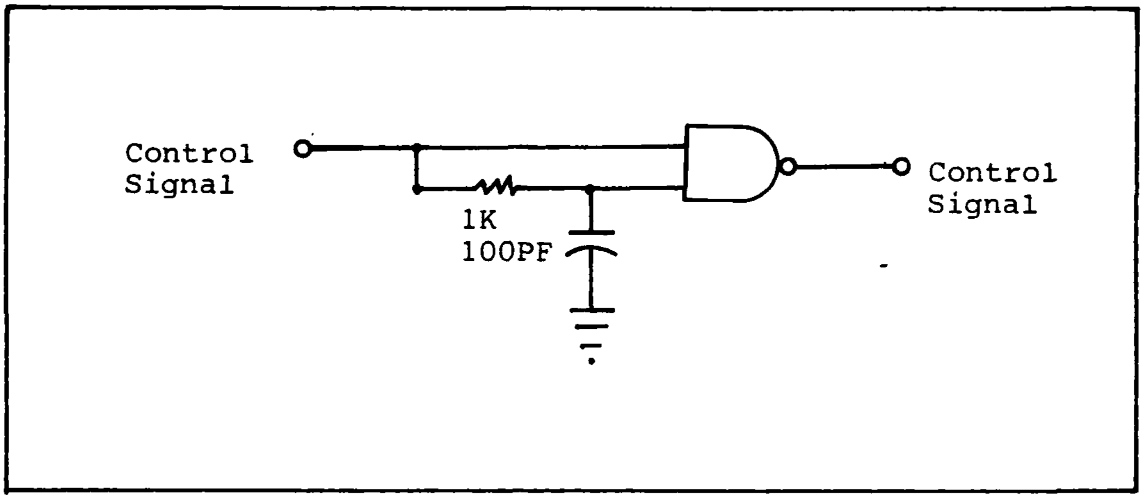


Fig. 8 Filter Circuit

(Ref 2:A13)

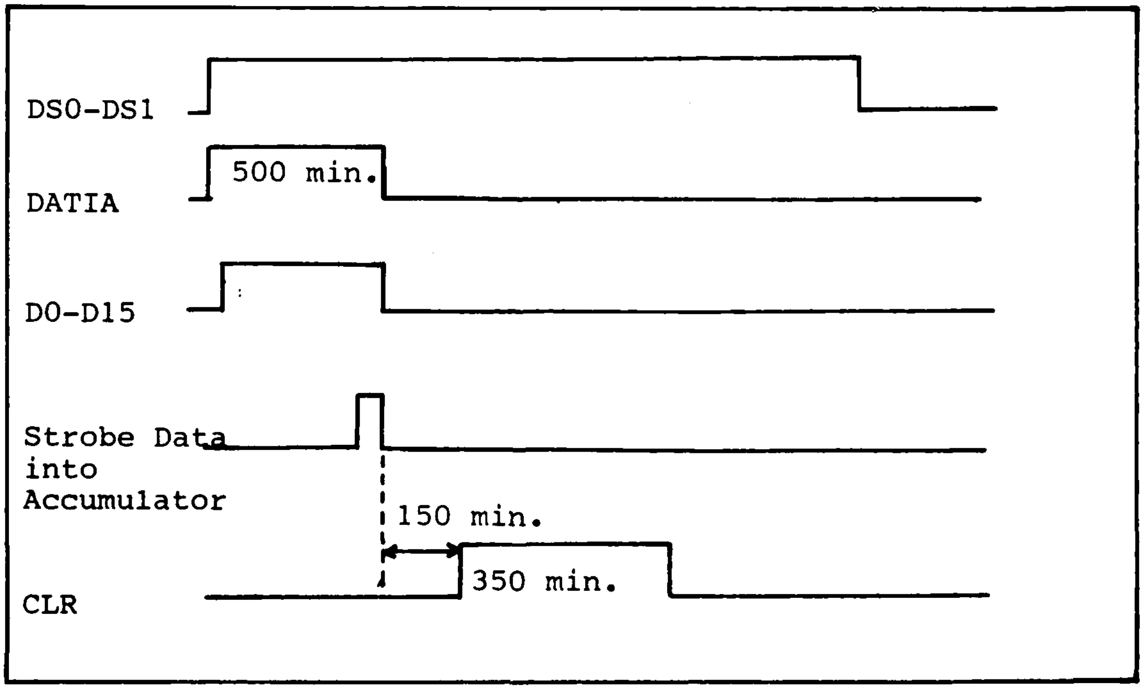


Fig. 9 Signal Timing Diagram

IV Specification of the Interface

This project consists of designing an interface that will connect a Talos parallel digitizer to the NOVA 1200 computer. The interface must meet the following specifications:

1. The inputs to the interface will be a 16-bit parallel data word from the digitizer, and a clear bit and DATIA signal from the NOVA 1200 computer.
2. The output from the interface to the computer will be a 16-bit parallel data word, a busy bit, and a done bit.
3. The 16-bit parallel data word from the digitizer must be converted to serial form and transmitted according to the RS-232C specification. At the receiving interface the serial data will be converted back to parallel form.
4. The interface must be as electrically isolated as possible from the digitizer and the NOVA 1200 computer.

The interface block diagram is shown in Figure 10. The interface consists of two major sections. Section I converts the 16-bit parallel data word output of the digitizer into two 8-bit serial words and transmits the serial data words along a RS-232C line. The format of the serial data words is shown in Figure 11.

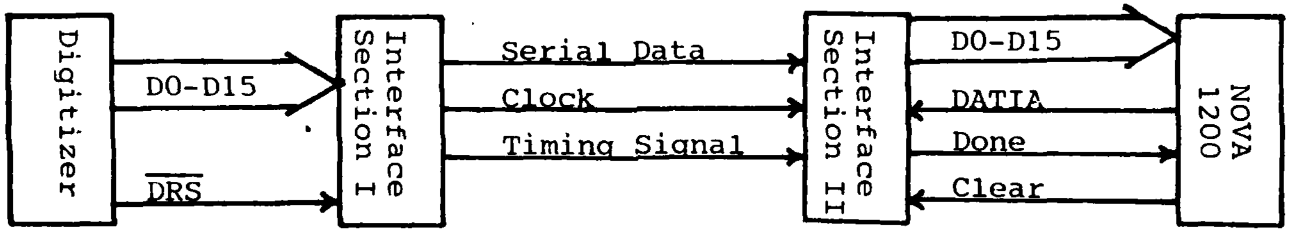


Fig. 10 Interface Block Diagram

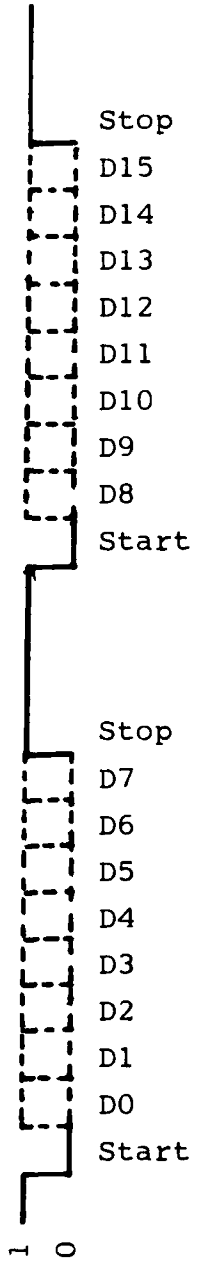


Fig. 11 Format of Serial Words

Section II of the interface receives the two consecutive serial data words and converts them into one 16-bit data word. Section II also generates a done bit that enables the computer to load the parallel data word into the accumulator at the proper time. Section II is physically located inside the cabinet of the computer.

The interface is enabled only when addressed by the computer. The address assigned to the interface is 44 octal. The computer places the address lines in the high state from the time the done bit is first tested until the parallel data word is loaded into the accumulator. The done bit is then cleared by the computer. A busy bit is also provided by the interface that may be needed if the digitizer is connected to a different computer in the future.

Electrical isolation of all signal lines to and from the interface is achieved utilizing an opto-isolator circuit.

Section I of the Interface

Section I of the interface is a parallel to serial converter. The block diagram of section I is shown in Figure 12. The primary component is an AY1013 Universal Asyncronous Receiver Transmitter (UART). This particular UART was chosen because it was readily available (on bench stock) and has separate input/output lines and control lines.

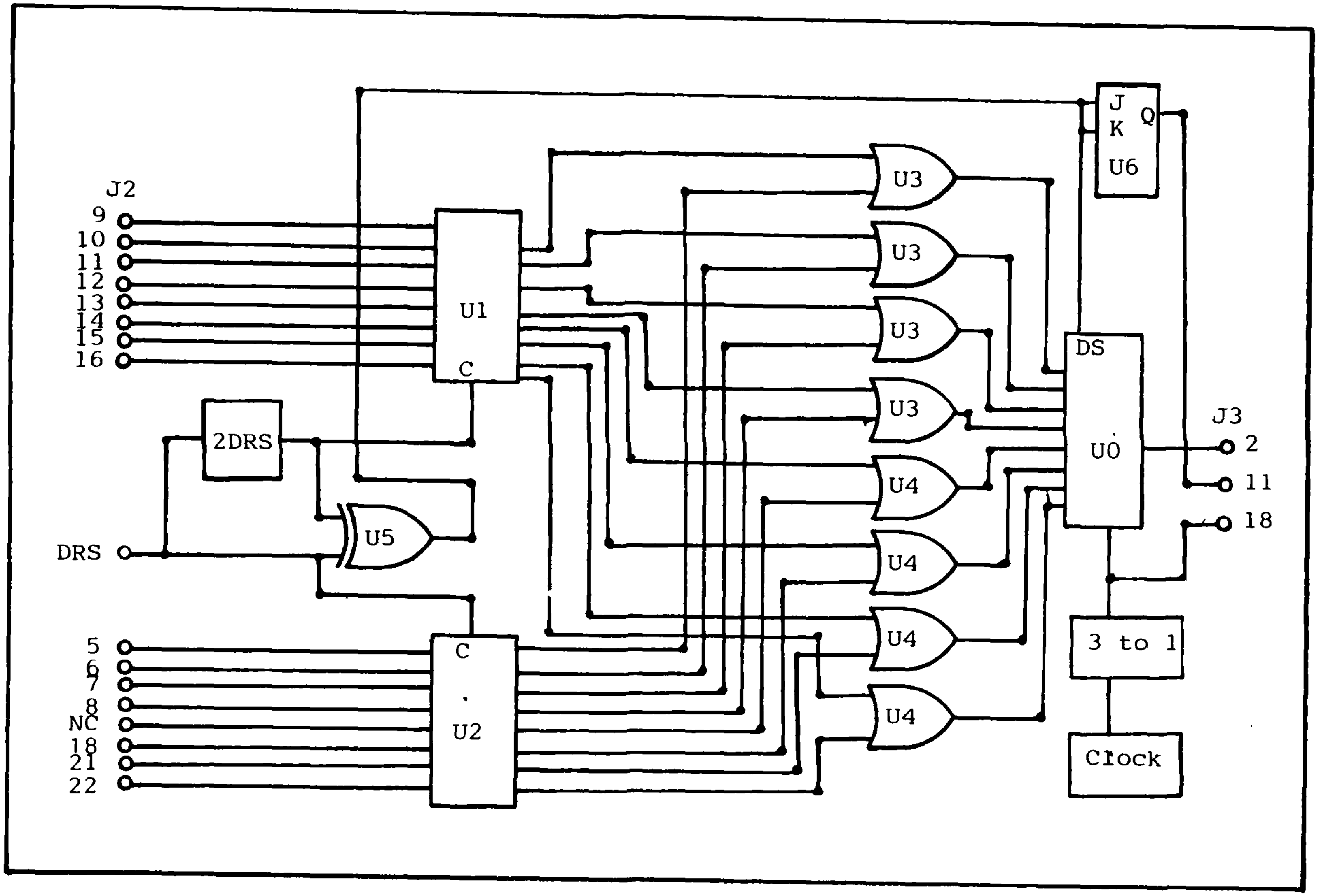


Fig. 12 Block Diagram of Section I of the Interface

Table VI

Uart Control Word Input

Pin No.	Name	Input Voltage
34	Control Strobe	+5 V.
35	No Parity	+5 V.
36	Number of Stop Bits	0 V.
37	Number of Bits	+5 V.
38	Number of Bits	+5 V.

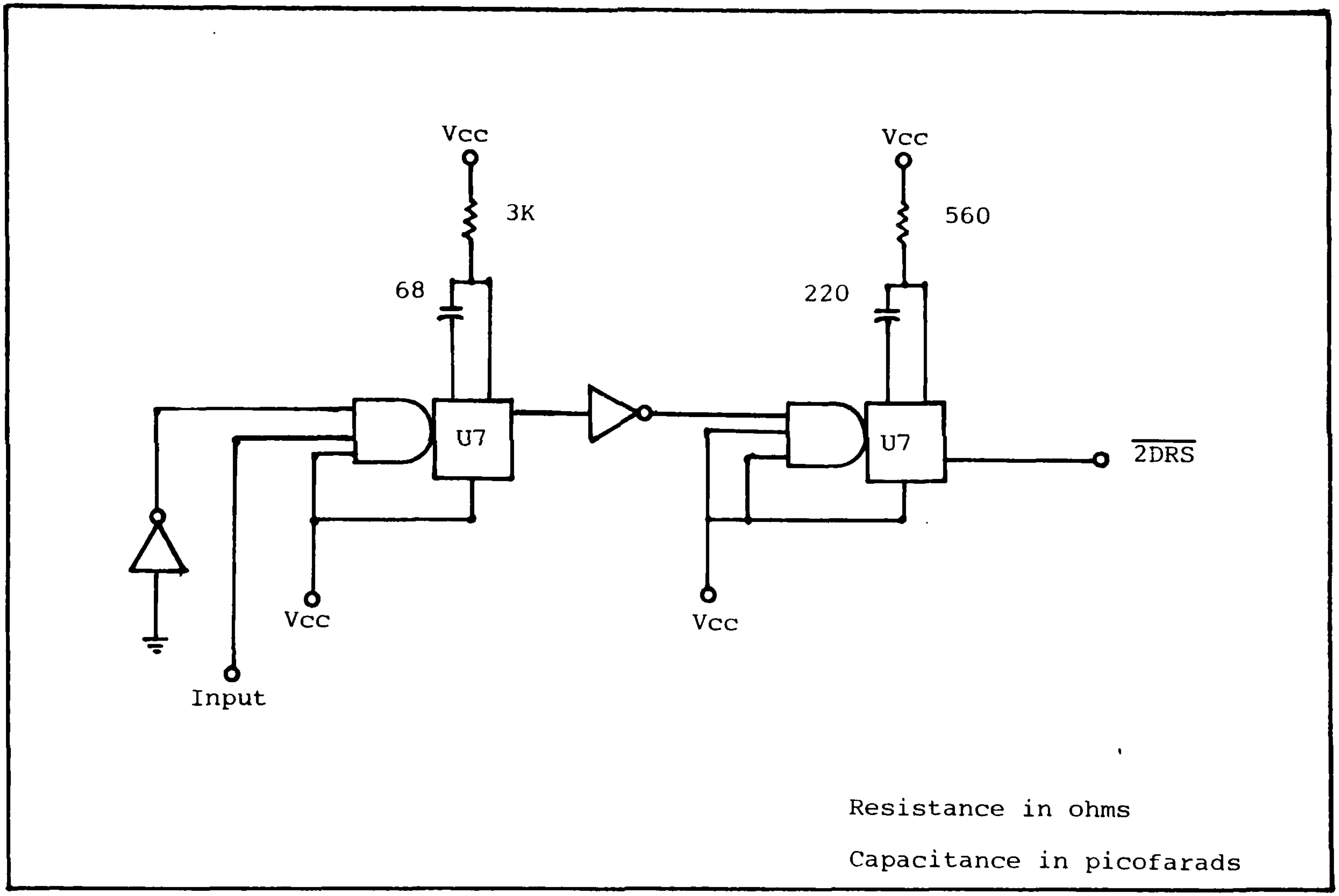
This feature is desirable as it enables the control lines to be hardwired. The control lines were hardwired as shown in Table VI. Wiring the control lines in this manner sets the UART for no parity, 8 data bits per character, 1 start bit, and 1 stop bit.

The 16 bit parallel data words must be separated into two eight bit words and then transmitted serially along a standard RS-232C data line. Two SN74116 dual 4-bit latches are used to separate the data into two 8-bit parallel words. The first eight bits (D0-D7) of the data word are latched onto the input of the UART and transmitted serially. The next eight bits (D8-D15) are then latched onto the inputs of the UART and transmitted.

A DRS pulse strobe occurs upon the arrival of a 16-bit parallel data word from the digitizer. The DRS pulse is shifted to latch bits D8-D15 after data bits D0-D7 have been transmitted by the UART. The diagram of the DRS shifter is shown in Figure 13. The majority of this circuit is contained on a SN74123 Dual Retriggerable Monostable Multivibrators with clear. The resistors and capacitors in the circuit form a time constant that determines the amount the pulse is shifted. The shifted DRS pulse and the original DRS pulse are used as an input to an Exclusive-OR gate. The output of this gate is a pulse at twice the frequency of DRS. This pulse will be designated 2DRS. The 2DRS pulse strobes the UART.

The 2DRS pulse is also an input to a JK flip-flop (U4) that generates a timing pulse that is used in Section II of the interface.

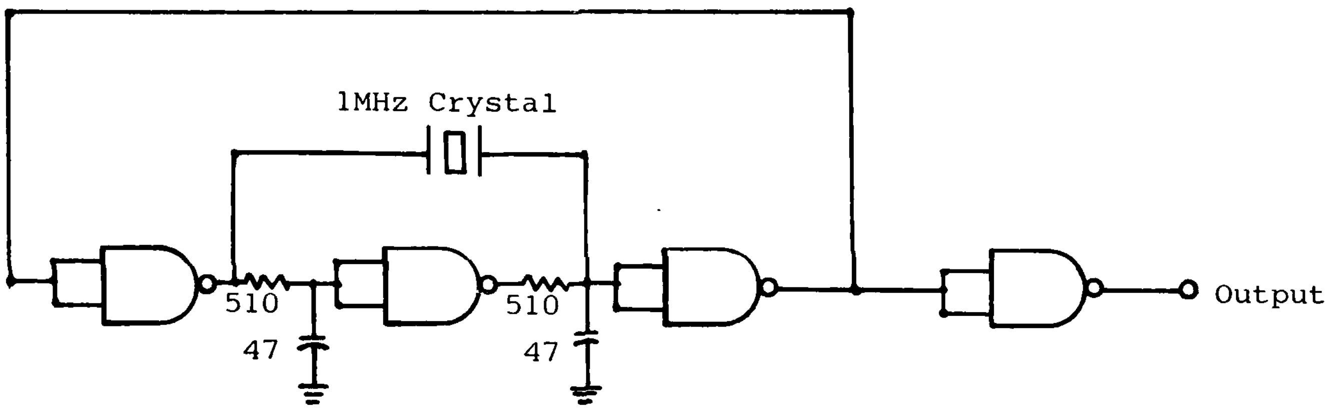
A clock pulse is required by both the transmitting and receiving UARTs. The clock pulse is generated in Section I and is sent via a RS-232C line to Section II. The diagram of the circuit that generates the clock pulse is shown in Figure 14. The output frequency of the clock circuit is approximately 1 MHz. The clock frequency is counted down to a frequency compatible with the UART. A 3 to 1 counter is used to count the frequency down to 333 kHz. The counter circuit block diagram is shown in Figure 15.



Resistance in ohms

Capacitance in picofarads

Fig. 13   DRS Pulse Shifter (Ref 3 :404)



Resistance in ohms

Capacitance in picifarads

Fig. 14 1MHZ Crystal Oscillator (Ref 3 :693)

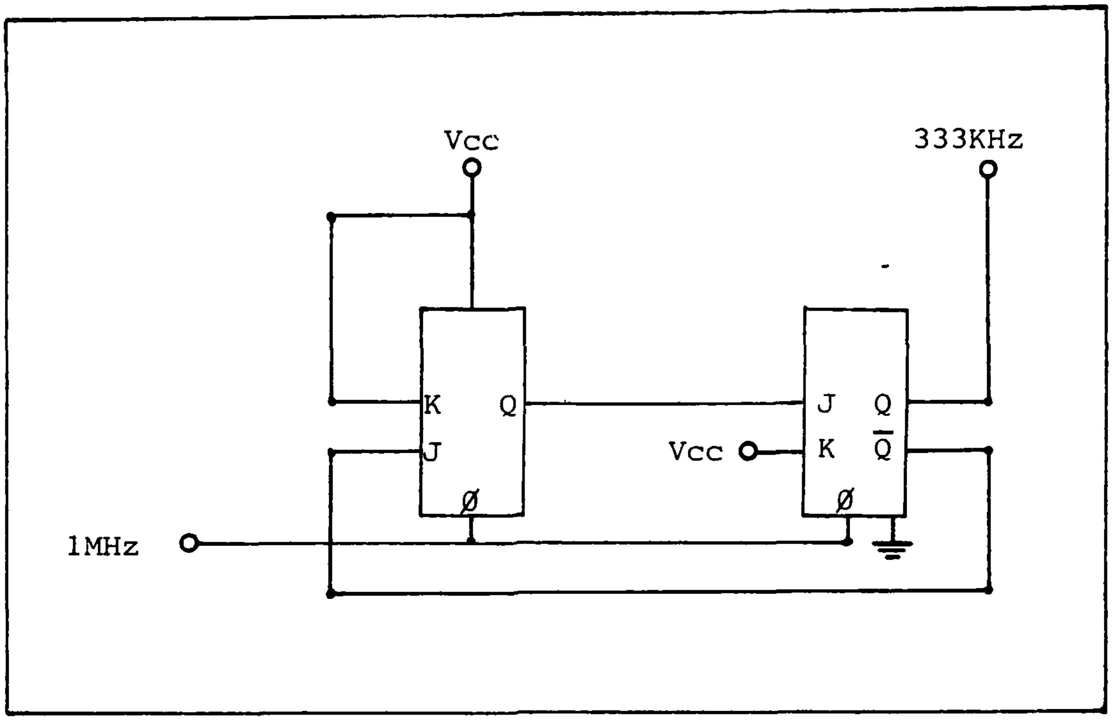


Fig. 15 3 to 1 Counter (Ref 4:76)

RS-232C

The three signals transmitted from Section I of the interface to Section II are the serial data, the clock signal, and a timing signal. These three signals are transmitted in accordance with the Electronics Industries Association (EIA) Standard RS-232C. The signals that are sent and the pin numbers assigned are shown in Table VII. The electrical specifications of the standard are met by using a Motorola MC1488 and MC1489, RS-232C line driver and line receiver respectively. The line driver converts the signals to be transmitted to +12 V maximum levels. At the receiving interface the line receiver converts the signals back to +5 V maximums.

Table VII

RS-232C Specification Pin Connections

Pin No.	Name
1	Protective Gnd
2	Serial Data
7	Signal Gnd
11	Timing Signal
15	Clock

Section II of the Interface

The two major functions performed by Section II of the interface is to convert serial data back to parallel data, and to perform the tasks necessary to load data into the computer. The block diagram of Section II is shown in Figure 16.

The receiving UART converts the serial input into a 8-bit parallel word. The parallel data word is loaded into the inputs of two SN74116 latches. One SN74116 is triggered to latch an 8-bit data word corresponding to D0-D7, and the second SN74116 is triggered to latch D8-D15. The UART generates a Data Available (DAV) pulse after each serial data word has been converted to a parallel word. The DAV pulse is used as an input to the circuit shown in Figure 17 to

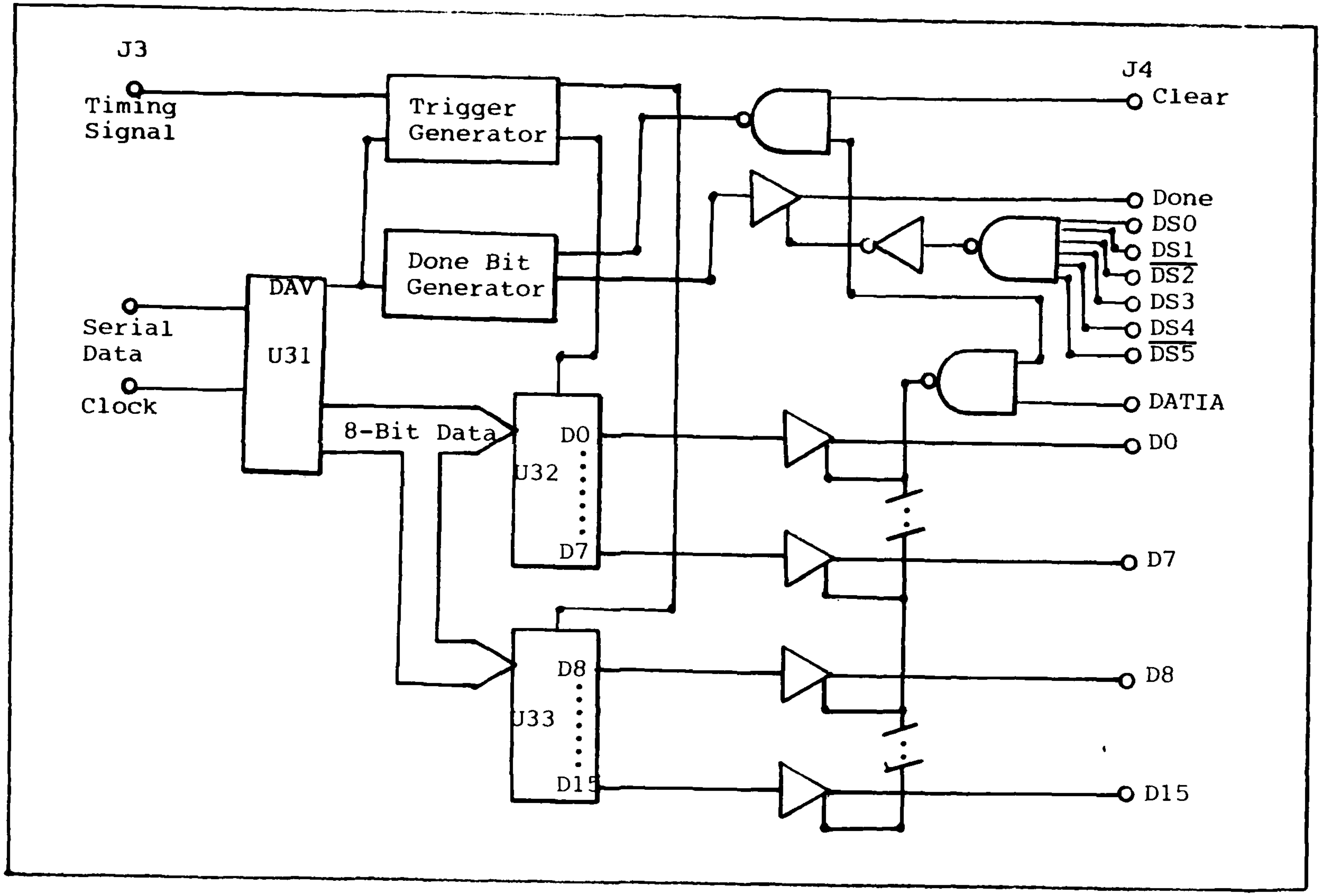


Fig. 16 Block Diagram of Section II of the Interface

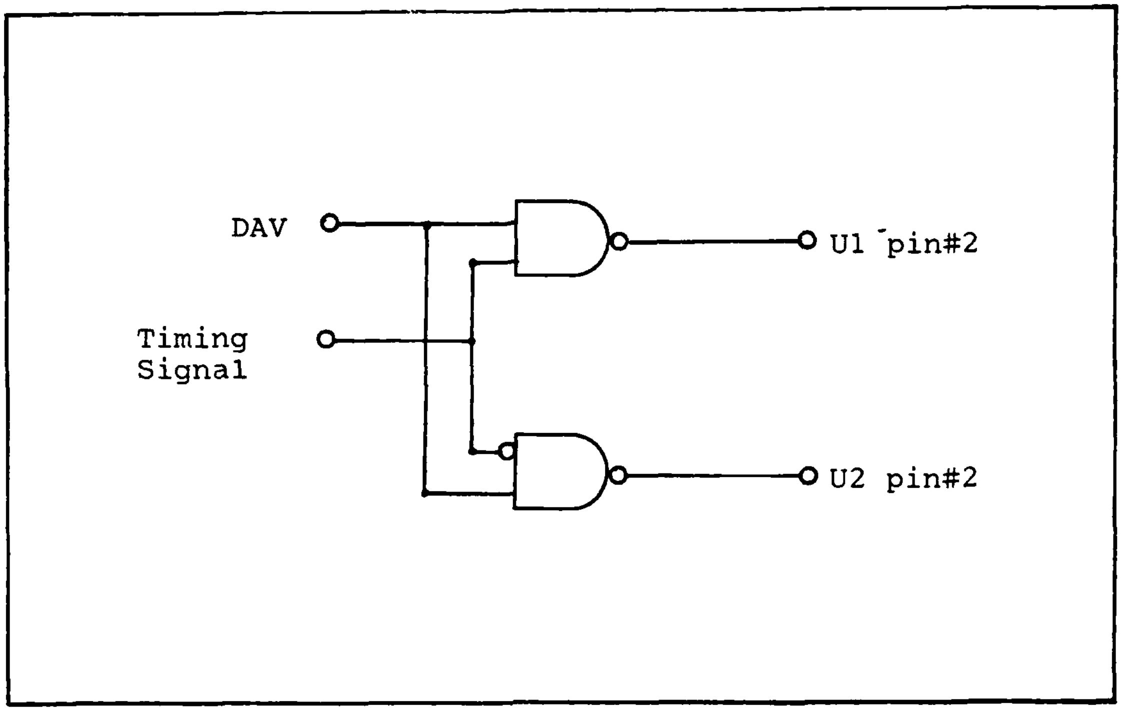


Fig. 17 Trigger Circuit

obtain the triggers for the two latches. The other input to the circuit is the timing signal received from Section I of the interface. The levels of the timing signal correspond to the transmission of the D0-D7 data word or the D8-D15 data word by Section I.

The output of the SN74116 latches is the input to the SN74LS367 Hex 3-State Bus Drivers. The bus drivers isolate the data output from the interface from the computer until the interface is addressed by the computer and the DATIA pulse goes to the high state.

To initiate action the computer addresses the interface on the data lines DS0-DS5. The address lines are decoded by a SN7430 eight input positive-NAND gate. The output of the NAND gate enables the interface to communicate with the computer and to load data onto the computer data lines. The address of the interface device is 44 octal. The address of the device is held on the address lines for the durarion of the input cycle.

The computer clears the done bit to initialize the interface device. When the 16-bit parallel data word is ready to be loaded into the accumulator of the computer the interface device sets the done bit to logic 1 (+5 V). The 16-bit data word is then strobed into the accumulator from the data lines. The computer then clears the interface device and continues with its' program until a new data word is needed.

The maximum time required for the computer to complete an instruction is 4 microseconds (Ref 2:D12), with the average time being approximately 2 microseconds. The digitizer has a 16-bit data word ready for output every 5 milliseconds. Taking the worst case the computer has time to execute 1250 instructions between input operations. This is important as the digitizer has a data word ready for input every 5 milliseconds regardless if it is addressed by the computer or not.

The done bit generator block diagram is shown in Figure 18.

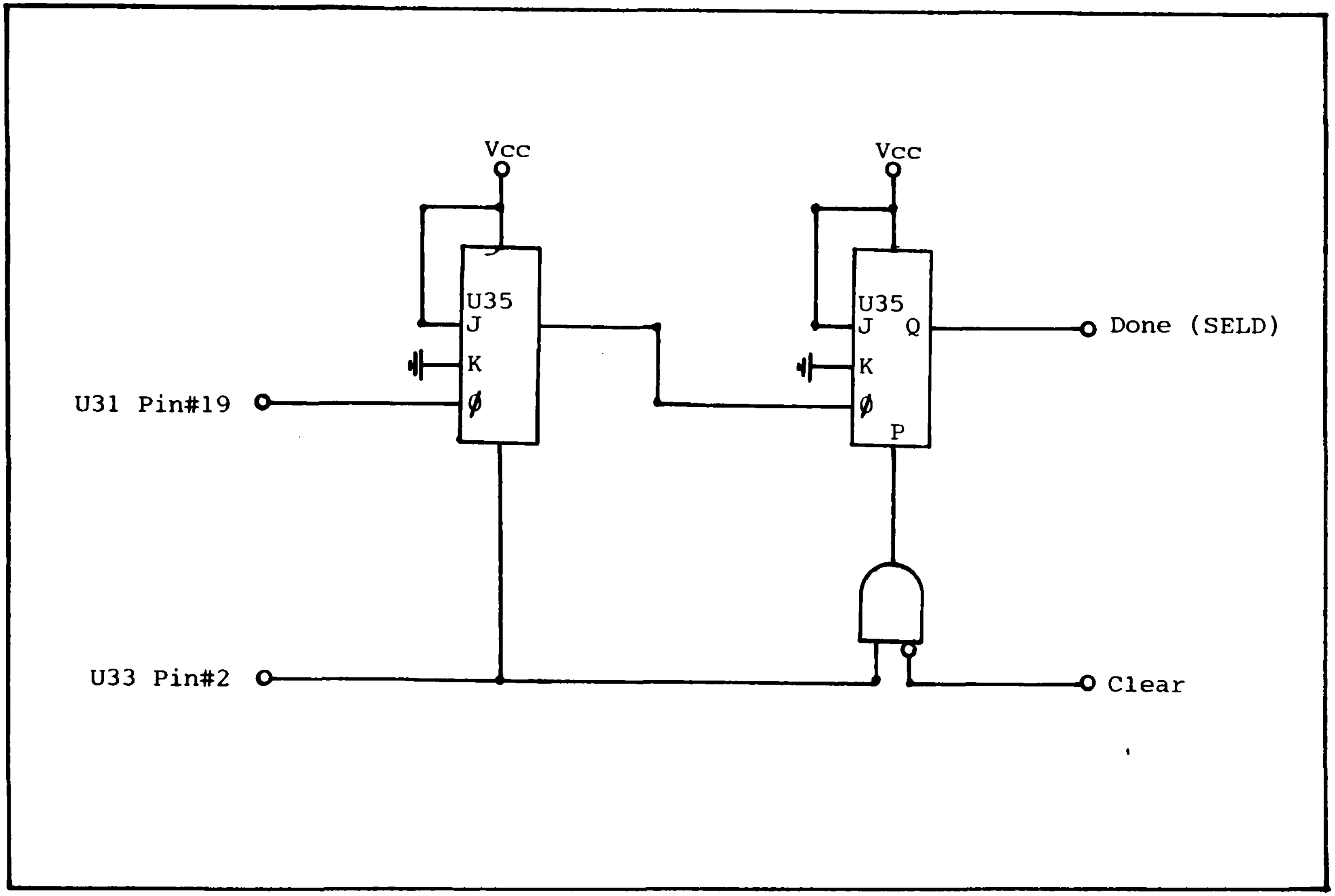


Fig. 18 Done Bit Generator

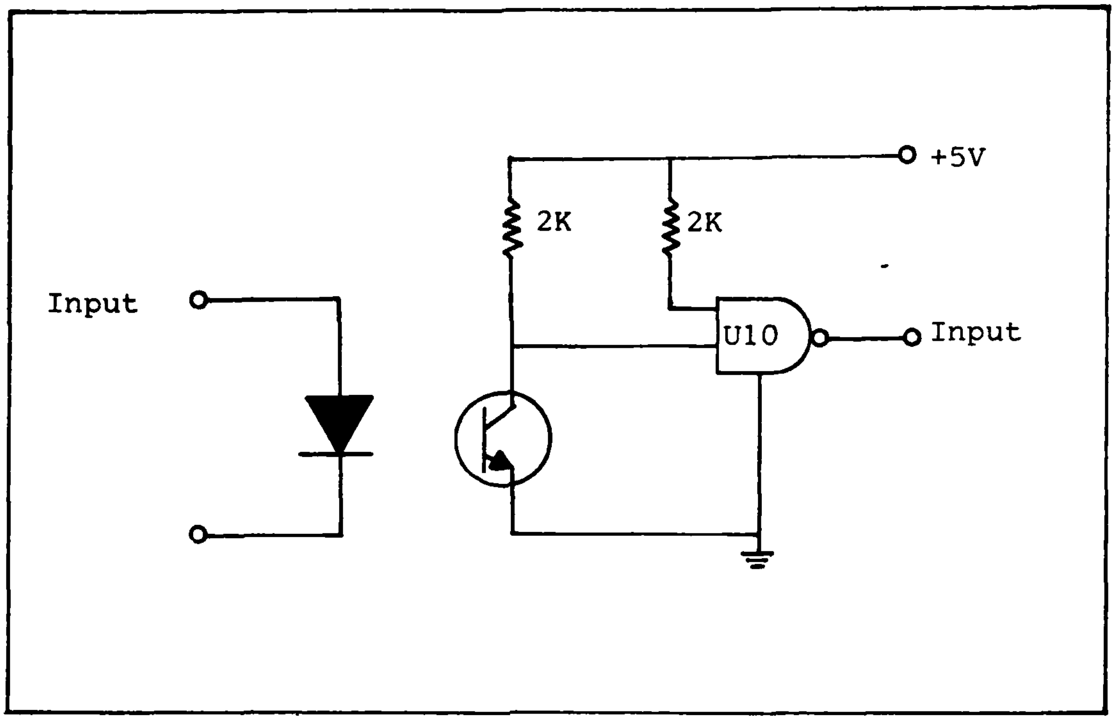


Fig. 19 Optoisolator Circuit (Ref 9 :139)

A done bit is generated after each pair of DAV pulses, signifying that a 16-bit data word is available for transfer. The timing signal connected to the clear input of the first flip-flop insures that the done bit is set to logic 1 after the second eight bit word (D8-D15) is available. The second flip-flop in the circuit allows the done bit generator to be cleared.

Electrical Isolation

Electrical isolation can be achieved using optoisolators on all input and output lines of the interface device. The optoisolator used was the Texas Instrument TIL-111. The optoisolator circuit used is shown in Figure 19.

The SN74132 is a Schmitt triggered NAND gate that is required to enable the TIL-111 to drive TTL loads.

To obtain electrical isolation the ground and power supply on the input of the optoisolator must be different from the ground and power supply on the output side of the isolator. This means that the computer would have to supply power to the SN74132 integrated circuit packages to achieve true electrical isolation. The computer does not provide a power supply output. There is less likelyhood of damage to the computer if the SN74132 circuit packages are supplied power by the interface. Although this does not provide isolation in the strictest sense it does provide a degree of protection.

Power Supply

The power supply requirements for Section I of the interface are +5, +12, and -12 volts. The +5 volts is required to power the TTL devices and the UART. The UART also requires a -12 volt power supply. The +12 volt power supply is required by the digitizer. The digitizer power is routed through the J1 connector. The digitizer draws too much current to be powered by the power supply contained in Section I. The power supply circuit diagram is shown in Figure 20. The power supply is rated at 450 milliampres of current. Section I of the interface draws approximately 400 milliampres of current. The digitizer is rated at 250 milliampres of current consumption but on the average

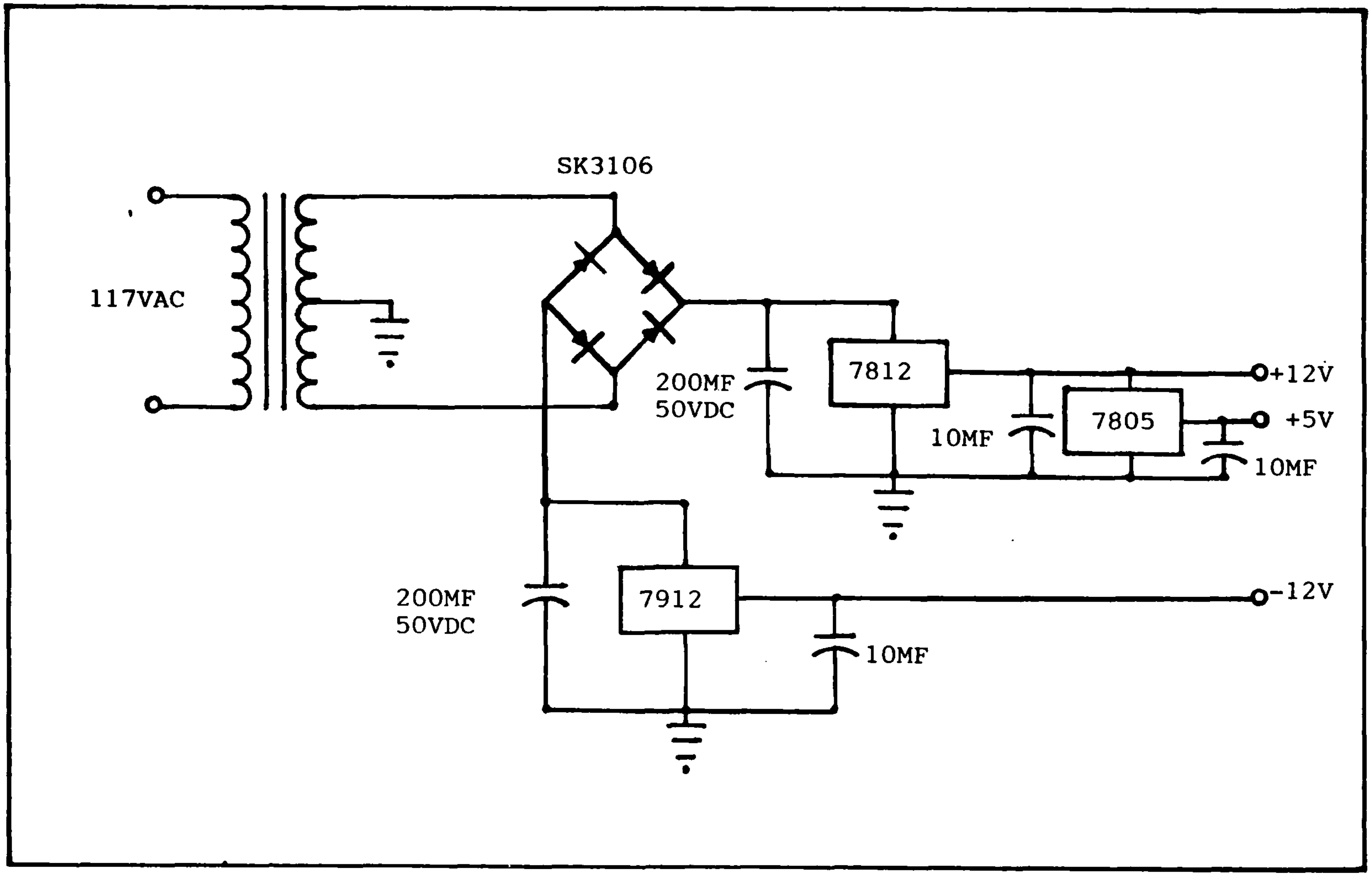


Fig. 20 Power Supply Circuit Diagram

uses approximately 120 milliampres. External power jacks for +12 volts are provided on the back of Section I to provide power for the digitizer. Section II of the interface requires +5 and +12 volts power supplies. Section II draws 450 milliampres of current. A larger transformer was used in the power supply of Section II to provide this larger amount of current. The transformer used will provide 850 milliampres of current.

V Software

Software is required to load the data from the digitizer into the accumulator of the computer. The data can then be used in various grid intersect coding algorithms. After the data has been manipulated by the grid intersect algorithms it can be used as the output to a plotter. The flow chart of the data loading program is shown in Figure 21.

The accumulators used in this program are accumulators 0 and 1. Memory locations 000177_8, 000176_8, 000175_8, and 000174_8 must be initialized prior to program execution. Memory location 000177_8 is initialized with the number of data words that are to be loaded into the computer. Memory location 000176_8 is initialized with the address of the memory location at which the first data word loaded is to be stored. This address will be designated Addr. Memory location 000175_8 is initialized with the number Addr-1. Memory location 000174_8 is loaded with the number Addr-2. The following are the steps of the program.

1. Clear the done bit generator.
2. Test the done bit until it goes to logic 1.
3. Load the data word into Accumulator 0.
4. Check the data word to determine if it is the same data word loaded in the previous data transfer. If it is the same data word return to Step 2.

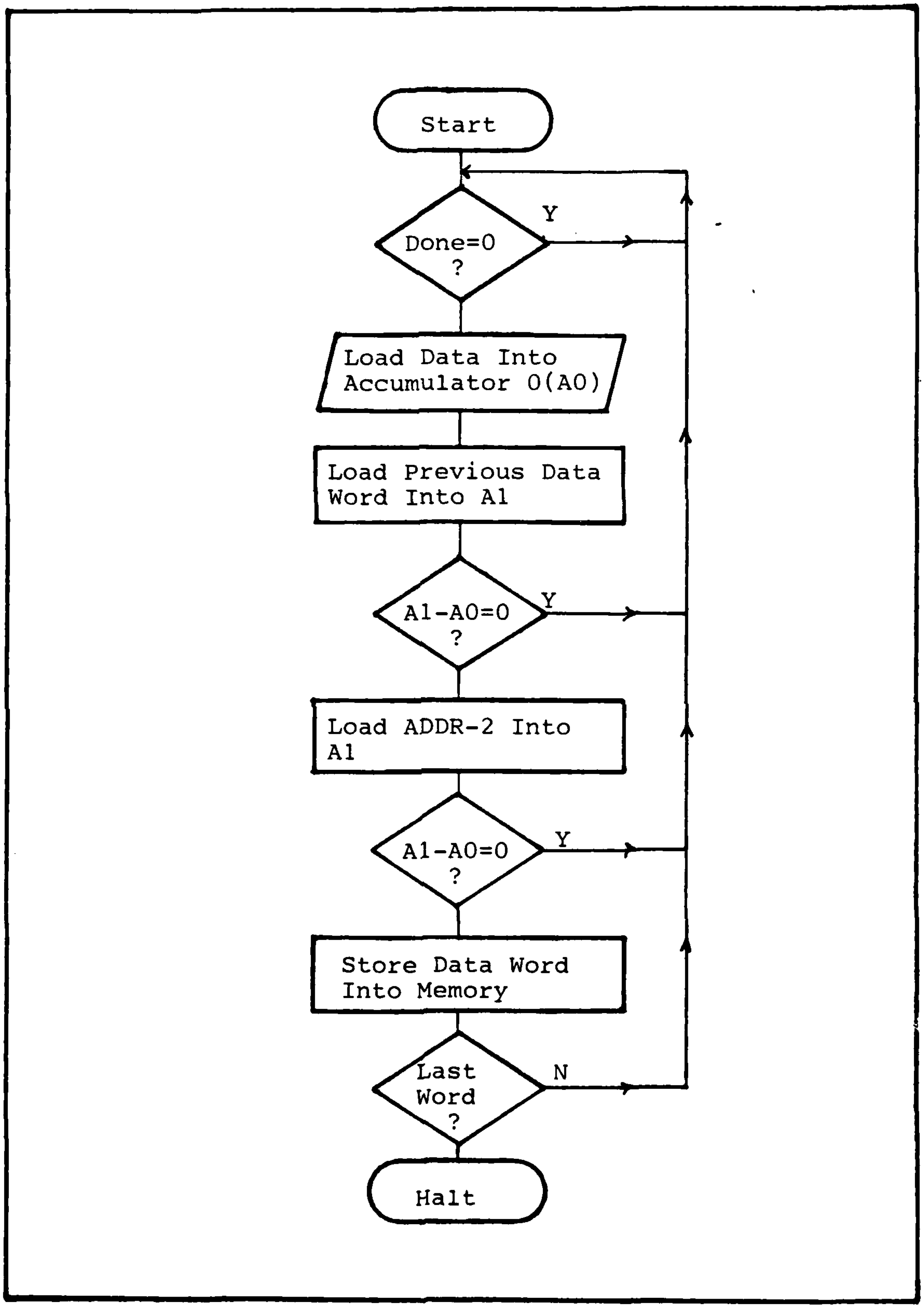

Start
Done=0 ?
Y
Load Data Into Accumulator 0(A0)
Load Previous Data Word Into A1
A1-A0=0 ?
Y
Load ADDR-2 Into A1
A1-A0=0 ?
Y
Store Data Word Into Memory
Last Word ?
N
Halt

Fig. 21 Loader Program Flow Chart

5. Check the data word to determine if it is the same data word that was loaded two data transfers ago. If it is the same data word return to Step 2.

6. Store the contents of Accumulator 0 in memory. Use the contents of memory location 000176_8 as the relative address of the location at which to store the data word.

7. Increment contents of memory location 000176_8.

8. Increment contents of memory location 000175_8.

9. Increment contents of memory location 000174_8.

10. Decrement contents of memory location 000177_8.

11. If the contents of memory location 000177_8 does not equal zero then return to Step 2.

12. Halt.

The digitizer has an output of a X-coordinate and then a Y-coordinate. To assure that the same coordinates are not loaded consecutively, the data word is compared to both of the previous two data words loaded into memory. This is accomplished by program steps 4 and 5 above. Loading the same coordinate pairs consecutively would not alter the representation of the line quantized but would be redundant information and would waste memory. The program to load the data from the digitizer into the computer is listed in Appendix B.

VI Conclusions

The basic objective of this thesis was to develop a system that could be used to evaluate various grid intersect codes. Although this system was developed it is cumbersome to work with. The NOVA 1200 computer is adequate for use in this system but peripherals need to be added to it to give the user more flexibility and ease of operation. The addition of a disk based operating system and the associated software would make this system more user oriented and increase its capability to handle complex algorithms. The digitizer itself is not very reliable. The digitizer was purchased in kit form and it is not known if the problems are from assembly of the kit or inherent in the design. One of the problems encountered is that two of the bits that are output from the digitizer are stuck in the high state. This was a problem that developed while testing the interface. Another problem encountered was erratic output of the digitizer. The pen could be put at a specific coordinate point on the grid surface and the output noted. If the pen was lifted from the grid surface and then replaced at that same grid coordinate the output from the digitizer would be sigificantly different. To make this a usable system another digitizer should be used.

Bibliography

1. Data General 093-000062-01. Introduction to Programming the NOVA Computers. (Revision 01) Mass.:Data General Corporation. September 1972

2. English, William. Data General 015-000009-08. How to Use the NOVA Computers. (Revision 08) Mass.: Data General Corporation. June 1974

3. Markus, John. Modern Electronic Circuits Reference Manual. NY.: McGraw-Hill Inc. 1980

4. Mimms, Forrest M., III. Engineer's Notebook. "integrated Circuit Applications". TX.: Radio Shack, Tandy Corporation. 1980

5. RS-232-C EIA Standard. "Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange". Wash. DC.: Electronic Industries Association. August 1969

6. TALOS Systems 50101-1. DIGI-KIT-IZER. "Assembly Instructions: Basic Unit". Part No. 12027-2. AZ.: TALOS Systems Inc.

7. TALOS Systems 50102-1. DIGI-KIT-IZER. "Assembly Instructions: Parallel Binary Output Board". Part No. 11910-2. AZ.: TALOS Systems Inc.

8. Spencer, John D. and Dale E. Pippenger. The Voltage Regulator Handbook. TX.: Texas Instruments Inc. 1977

9. Texas Instruments. The Optoelectronics Data Book: for Design Engineers. (Fourth Edition) TX.: Texas Instruments Inc. 1976

10. Texas Instruments. The TTL Data Book for Design Engineers. (Second Edition) TX.: Texas Instruments Inc. 1976

Appendix A

Digitizer Interface Integrated Circuits

Reference Designation	Description	Part Number
U0, U31	UART	AY-1013A
U1, U2, U32, U33	Dual 4-bit Latches	SN74116
U3, U4	Two Inout NOR Gate	SN7402
U5	Two Inout Exclusive-OR Gate	SN7486
U6, U8	Dual J-K Flip-Flops with Clear	SN7473
U7	Dual Monostable Multivibrators	SN74LS123
U9, U34, U47, U48	Two Input Nand Gates	SN74LS00
U10, U11, U12, U13, U41, U42, U43, U44, U45, U46	Two Input Nand Schmitt Triggers	74LS132
U14-29, U50-68	Optoisolators	TIL-111
U30	RS-232C Line Driver	MC1488
U35	Dual J-K Flip-Flops with Preset and Clear	54LS109
U36, U37, U38	Hex Bus Drivers	SN54367
U39	8-Input Nand Gate	SN7430
U40	2-Input And Gates	SN7408
U49	RS-232C Line Receiver	MC1489

Appendix B

Loader Program

Following is the data loading program to drive the digitizer interface.

;ROUTINE TO LOAD DATA WORDS FROM THE DIGITIZER

;INTO THE COMPUTER. EACH CHARACTER IS CHECKED

;TO INSURE THAT THE SAME COORDINATE PAIR IS

;NOT LOADED CONSECUTIVELY.

;ACCUMULATORS USED ARE 0 AND 1.

;INITIALIZE MEMORY LOCATIONS:

; 000177_8 TO NUMBER OF DATA WORDS TO BE

;   LOADED.

; 000176_8 TO ADDRESS AT WHICH FIRST DATA

;   WORD IS TO BE LOADED (ADDR).

; 000177_8 TO ADDR-1.

; 000174_8 TO ADDR-2.

```
060244    NIOC    DIG      ;CLEAR DEVICE
063644    SKPDN   DIG      ;TEST DONE BIT
000777    JMP     .-1      ;
060544    DIAS    0,DIG    ;LOAD DATA INTO ACCUMULATOR 0
026175    LDA     1,@174   ;LOAD PREVIOUS DATA WORD
106415    SUB     0,1      ;CHECK IF TWO DATA WORDS ARE SAME
000773    JMP     .-5      ;
```

```
026174    LDA    1,@174  ;LOAD PREVIOUS DATA WORD-1
106415    SUB    0,1     ;CHECK IF TWO DATA WORDS ARE SAME
000770    JMP    .-8     ;
042176    STA    0,@176  ;STORE DATA WORD
010176    ISZ    176     ;INCREMENT CONTENTS OF 000176$_8$
010175    ISZ    175     ;INCREMENT CONTENTS OF 000175$_8$
010174    ISZ    174     ;INCREMENT CONTENTS OF 000174$_8$
014177    DSZ    177     ;DECREMENT CONTENTS OF 000177$_8$
000761    JMP    .-14    ;
063077    HALT
```

Vita

Jerry N. Peery was born on 20 Oct 1953 in Cortez, Colorado. He graduated from San Juan High School, Blanding, Utah in May 1972. In February 1973 he enlisted in the United States Air Force as an Air Traffic Control Radar Repairman. He was honorably discharged from the Air Force in December 1976. In May 1981, he graduated from the University of Oklahoma with a Bachelor of Science degree in Electrical Engineering and was commissioned throught AFROTC at the same time. He entered the Air Force Institute of Technology in June 1981 as his initial assignment.

Permanent Address: P.O. Box 60
Blanding, Utah